UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

LES SYSTÈMES EXPERTS ET LA RECHERCHE OPÉRATIONNELLE

par

Martin Desrochers

Université de Montréal
Centre de recherche sur les transports - Publication #386
Novembre 1984

SOMMAIRE

Les systèmes experts permettent de plus en plus d'intégrer des connaissances dans des sujets restreints et leur performance peut parfois dépasser celles d'experts humains dans ces domaines. Cet exposé tente de situer ce que sont les systèmes experts et de montrer les relations entre eux et la recherche opérationnelle: Comment les systèmes experts aident-ils et pourraient-ils aider la recherche opérationnelle et quelles techniques de la recherche opérationnelle pourraient aider les systèmes experts ? L'exposé se termine par un exemple d'application possible au routing.

TABLE DES MATIERES

Sommaire		1
Table des matières		ii
Introduction		1
1	Les systèmes experts	3
1.1	Les caractéristiques des systèmes experts	5
1.2	La représentation procédurale	12
1.3	La représentation déclarative	13
1.3.1	Les règles de production	14
1.3.2	Le calcul des prédicats	16
1.3.3	Les réseaux sémantiques	17
1.4	La représentation mixte	20
1.4.1	Les prototypes	21
2	Plus de détails	24
2.1	Sur les règles de production	25
2.2	Sur les prototypes	36
3	La recherche opérationnelle et les systèmes experts	42
3.1	Une méthode d'inférence	42
3.2	ALICE, un système expert	44
3.3	L'utilisation de la R.O.	46

4	Le problème de routing: un exemple d'application possible	49
4.1	Le design du système expert	52
4.2	Quelques éléments du savoir	54
4.3	Quelques détails importants	61
CONCLUSION		63
BIBLIOGRAPHIE		65

INTRODUCTION

Certains systèmes informatiques permettent d'effectuer, dans un champs de compétence restreint, des tâches habituellement exécutées par des experts. Ces systèmes sont appelés "SYSTEMES EXPERTS (S.E.)".

Ces systèmes ont pour caractéristiques générales de résoudre des problèmes, pour lesquels il n'existe pas d'algorithme de résolution, par des raisonnements s'apparentant à ceux d'un expert humain. Ils peuvent de plus expliquer les raisonnements ayant mené à la solution obtenue. Pour être utile, un tel système doit résoudre un problème avec un haut niveau de performance comparable à celui d'un expert humain, tout en communiquant avec l'usager dans un langage se rapprochant de sa langue naturelle.

Les S.E. permettent à un usager non-expert de résoudre des problèmes difficiles que seuls de rares experts peuvent solutionner efficacement. Un S.E. peut donc amener une économie de main-d'oeuvre considérable.

La première section de cet exposé décrira les systèmes experts, leurs composantes. L'accent sera mis sur les modes de représentation des connaissances et les modes de contrôle du raisonnement. Trois types de modèles de représentation du savoir seront exposés.

La seconde section examinera plus en détail deux des modèles de représentation du savoir qui présentent un intérêt particulier pour la recherche opérationnelle: les règles de production et les prototypes.

La troisième et dernière section fera le point sur les rapports entre la recherche opérationnelle et les systèmes experts, tant au point de vue de l'utilisation des techniques R.O. dans les systèmes experts que des systèmes experts traitant des problèmes R.O.; nous pourrons voir que les relations sont faibles actuellement. L'exposé se terminera par une proposition de développement d'un système expert appliqué à un domaine de la R.O.: le routing.

SECTION 1

Les systèmes experts

Depuis le début des années 70 , de nombreux systèmes experts ont été développés dans différents domaines d'application. Plusieurs systèmes visaient des applications médicales diverses, soit MYCIN , pour le diagnostic et le traitement des maladies infectieuses du sang, et, CADUCEUS, pour le diagnostic des maladies internes. Tous ces systèmes sont à un stade plus ou moins expérimental et ont un niveau de performance relativement bon.

D'autres domaines d'application ont permis le développement du S.E., qui surpassent clairement la performance des meilleurs experts humains et qui sont utilisés régulièrement. Le système DENDRAL analyse des données de spectographie de masse et d'autres expériences chimiques pour déterminer la structure moléculaire de composés organiques. Un système complémentaire, META-DENDRAL, est utilisé pour ajouter des connaissances d'analyse à DENDRAL.

Un autre système utilisé de façon courante, MACSYMA, effectue le calcul intégral et différentiel de façon symbolique et excelle à la simplification d'expressions mathématiques. Le système R1, quant à lui, planifie la configuration des systèmes d'ordinateurs DEC VAX, de façon à répondre aux spécifications du client.

Tous ces S.E. ayant des niveaux de performance variant de bon à excellent partagent la particularité d'avoir un champ de compétence étroit et, où peu d'experts humains excellent. Les tentatives de résoudre d'autres problèmes simples que les humains accomplissent journalièrement ont résulté en des taux de performance plutôt décevants. En effet, un des meilleurs S.E. permettant la reconnaissance de la parole, HEARSAY-II, est capable de comprendre un discours cohérent avec l'habilité d'un enfant de 10 ans et un vocabulaire anglais d'environ 1000 mots.

Les S.E. peuvent être envisagés raisonnablement pour résoudre plusieurs types d'activités:

1) l'analyse et l'interprétation de données pour leur trouver une signification. Un exemple d'un tel système est DENDRAL qui trouve la structure moléculaire d'un composé organique à l'aide de données d'une spectographie de masse;

2) le diagnostic et le traitement des défauts dans un système. MYCIN, par exemple, diagnostique et trouve le traitement pour les maladies infectieuses du sang;

3) le contrôle de processus en temps réel. Le système doit interpréter des signaux et envoyer une rétroaction au processus, par exemple, le contrôle d'un respirateur chez un patient en choc post-opératoire;

4) la prospective selon un modèle basé sur des données passées et présentes, par exemple, la prédiction des effets d'un changement de politique économique;

5) la planification d'une activité ou d'un système de façon à réaliser les objectifs, par exemple, planifier une expérience en génétique par MOLGEN;

6) le design d'objets ou de systèmes répondant à un ensemble de spécifications. Plusieurs équipes ont mis au point des systèmes faisant le design de circuits intégrés.

Tous ces domaines profitent des connaissances acquises dans le domaine de l'intelligence artificielle depuis vingt-cinq ans sur le raisonnement, et, qui sont maintenant appliquées de façon expérimentale pour le développement de S.E. Déjà certains S.E. sont très au point et nous pouvons prévoir que les S.E. connaîtront dans les prochaines années un développement considérable.

1.1 Les composantes des systèmes experts

Un système expert utilise une base de savoir constituée de faits, de stratégies et de règles heuristiques de raisonnement pour raisonner sur les données fournies interactivement par l'usager. Ce raisonnement est

contrôlé par un interprète inférant sur les connaissances et les données pour obtenir des conclusions, demander des données supplémentaires, répondre aux interrogations de l'usager sur le raisonnement ou sur l'utilité d'une information demandée par le système et même ajouter, des connaissances à sa base de savoir.

Pour réaliser le S.E. idéal décrit au paragraphe précédent, plusieurs composantes sont nécessaires: un interprète pour la langue naturelle, une base de savoir, un système cognitif, un espace de travail, un système justificateur, un système d'acquisition de connaissances. Dans les S.E. actuels, ces composantes n'existent pas toujours. Dans certains cas, certaines composantes sont manquantes (ex: interprète, système d'acquisition de connaissances,...) dans d'autres, les composantes sont fusionnées (ex: la base de savoir et le système cognitif).

L'interprète de langue naturelle sert à communiquer dans un sous-ensemble de la langue de l'usager humain. Les questions à l'usager sont souvent mémorisées, mais les réponses doivent être analysées et codifiées pour être intelligibles pour les autres composantes du S.E.

La base de savoir contient les connaissances d'un ou plusieurs experts humains sur un problème déterminé. Ces connaissances sont constituées de faits spécialisés (définitions, classifications d'objets, relations entre les objets, etc.), des jugements, de l'expérience et des heuristiques sur

le domaine du problème. Le S.E. pourra, en combinant les données de l'usager avec les connaissances de la base de savoir, tirer des conclusions et déterminer les données manquantes au raisonnement.

L'espace de travail contient l'état courant du système expert. Cet état peut se diviser en trois parties: l'agenda contient les tâches non-exécutées, la solution contient les valeurs des variables globales, les objectifs et les données. La dernière partie concerne le plan de raisonnement pour justification future et la stratégie utilisée pour résoudre cet exemplaire du problème.

Le système cognitif gère l'exécution des autres composantes du système expert. En général, cette gestion utilise les étapes suivantes:

1) choix de la prochaine activité à exécuter parmi les tâches de l'agenda;

2) réalisation de l'activité choisie. Ce peut être l'utilisation de connaissances pour tirer une conclusion, une demande d'information à l'usager ou une demande d'explication au système justificateur;

3) l'évaluation des probabilités pour les conclusions et des priorités pour les tâches de l'agenda. A la lumière de nouvelles informations, les conclusions changent de probabilités, certaines

deviennent plus vraisemblables et d'autres ne sont plus intéressantes, de même pour les tâches de l'agenda;

4) élagage des conclusions et tâches sans intérêts.
L'élimination de conclusions permet de concentrer la recherche sur un sous-ensemble de conclusions et d'accélérer l'obtention de la solution. Comme dans toute recherche dans un arbre, il s'agit d'une opération primordiale, car l'efficacité de la recherche dépend de l'élagage hatif des branches inutiles;

5) vérifier si l'objectif est atteint.
Le système cognitif est la partie intelligente du système. Il doit guider la recherche de façon à obtenir le plus tôt possible le maximum de conclusions à partir des données disponibles. Il doit aussi signifier à l'usager les informations qui permettraient d'obtenir le maximum de conclusions supplémentaires.

Le système justificateur produit, à partir de l'historique contenu dans l'espace de travail, la justification des conclusions obtenues et les raisons justifiant une demande d'information supplémentaire. Ceci permet à l'usager de mieux comprendre les conclusions et de décider de fournir ou non les informations demandées en connaissant les bénéfices à venir. Par exemple, si le S.E. hésitait entre deux maladies, et, qu'un test permettait de déterminer laquelle entre les deux, l'usager

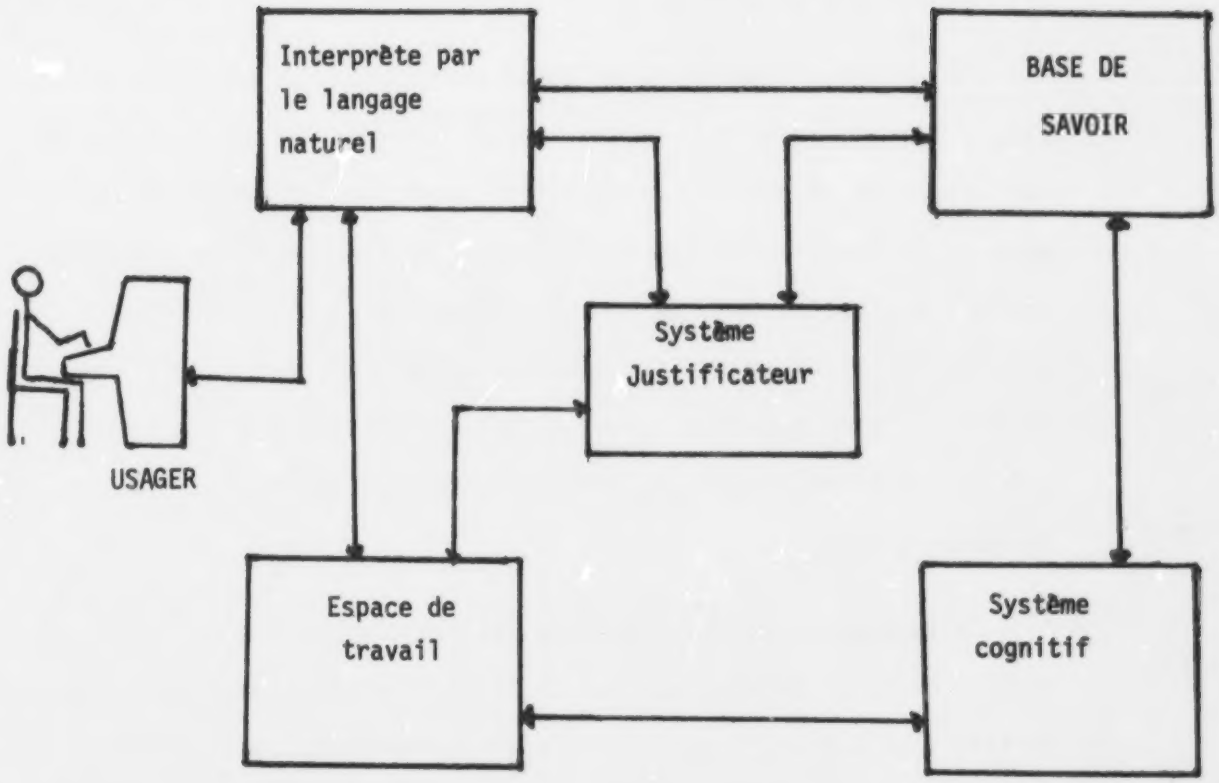

FIGURE 1

Communication dans un système expert

pourrait choisir de ne pas effectuer le test, si les deux maladies demandaient le même traitement et que la valeur marginale de l'information était presque nulle.

Le système d'acquisition de connaissances fait un peu bande à part. Il n'est accessible qu'aux experts qui s'en servent pour améliorer la base de savoir. La consultation permet à l'usager d'ajouter des morceaux de connaissances et de vérifier l'efficacité de ces connaissances. Le système doit, de son côté, s'assurer qu'il a toutes les informations nécessaires à son raisonnement et doit conserver l'intégrité de la base de savoir i.e., qu'il doit effectuer toutes les mises à jour qu'entraînent l'ajout ou la modification d'éléments dans la base de savoir.

Cette description fort générale des composantes d'un S.E. laisse place à beaucoup de questions sur la façon d'implanter les différentes parties d'un S.E. S'il est clair que la communication avec l'usager doit se faire dans la langue la plus naturelle possible, nous n'avons aucune indication sur la façon de représenter la base de savoir, la façon dont se fera l'inférence sur les connaissances et les données, etc. Tous ces problèmes ont de multiples solutions développées par les différentes écoles de l'intelligence artificielle. Deux étapes en particulier sont intéressantes, tant par l'intérêt des solutions développées que par le développement menant à ces solutions. Ce sont la représentation de la base de savoir et le système cognitif.

Ces deux étapes sont les plus déterminantes pour l'architecture d'un S.E. et elles sont étroitement reliées lors de l'implantation. Un S.E. fonctionnel demande que la méthode d'inférence et la représentation de la base de savoir soient compatibles. Dans cet exposé, les méthodes de représentation de la base de savoir seront exposées et accompagnées de la description de la méthode de raisonnement compatible, si nécessaire.

Le savoir dans un S.E. est fractionné en particules élémentaires (objets, relations, données, etc...). Ceci permet d'organiser le savoir selon une méthode de représentation, d'utiliser ces fragments pour déduire et expliquer les conclusions du système. La dimension des particules détermine l'efficacité du raisonnement, si les particules sont trop petites, la description du savoir sera très laborieuse et si elles sont trop grandes, certaines connaissances ne pourront être représentées fidèlement.

Deux approches principales de représentation des connaissances dans la base de savoir existent. La première, la représentation procédurale insiste sur le "savoir-faire" et décrit les connaissances comme étant imbriquées dans des procédures qui peuvent être appelées par différents mécanismes de contrôle. L'autre approche, la représentation déclarative insiste sur "connaître que" en séparant le contrôle du raisonnement et le savoir en tant que tel. Le savoir est représenté sous forme de prédicats,

de règles de production ou de réseaux sémantiques. Depuis quelques temps, des chercheurs ont combiné ces deux approches pour obtenir une représentation mixte du savoir dont un exemple sera présenté.

1.2 La représentation procédurale

Les automates finis sont l'une des deux représentations procédurales du savoir. Ils peuvent être utilisés pour représenter les actions à accomplir, les états représentant les actions et les arcs représentant les données, les règles pour passer d'un état à un autre. L'automate peut être déterministe, dans ce cas, le contrôle du raisonnement est très simple. Dans le cas non-déterministe, le mécanisme de contrôle est complexe, car on doit examiner tous les chemins partant d'un état pour choisir celui qui aboutit au but désiré. Toutefois, le nombre d'états d'un automate non-déterministe est inférieur à celui de l'automate déterministe effectuant la même tâche.

La seconde représentation procédurale possible est d'imbriquer le savoir dans des procédures. Le contrôle du raisonnement peut être effectué à l'aide d'appel direct si le raisonnement est entièrement déterministe. Le contrôle peut aussi être confié à des démons. Un démon DEMON(P,C) démarre l'exécution de la procédure P dès que la condition C est vérifiée, l'utilisation de démons demande au moniteur un effort considérable, car

il doit toujours vérifier si l'une des conditions est vérifiée. Une troisième façon de contrôler l'inférence est de déclencher l'exécution si le but à réaliser est celui que réalise la procédure. Chaque procédure possède un descripteur des buts réalisés. Le moniteur cherche parmi les procédures celle dont le but correspond le mieux au but à réaliser. Les procédures peuvent créer des sous-buts.

Dans l'approche procédurale, les interactions entre le contrôle et le savoir sont explicitement prédéterminées lors de la conception des procédures. Toutefois, l'ajout de nouvelles connaissances au savoir pose les mêmes problèmes que l'ajout d'une nouvelle procédure à un système complexe, i.e. des problèmes d'interactions entre les procédures qui doivent être réglées par un "expert" en informatique. Les changements au mécanisme de contrôle posent les mêmes problèmes.

1.3 La représentation déclarative

Le but premier de la représentation déclarative est de permettre de représenter les connaissances séparément du mécanisme de contrôle. Le système cognitif comporte un interprète analysant les connaissances et inférant sur les données et ces connaissances. Cette façon de procéder permet de construire des interpréteurs généraux de connaissances déclaratives plutôt qu'une procédure particulière pour chaque connaissance.

La représentation déclarative permet le développement de la base de savoir en modifiant uniquement celle-ci. Il n'est pas nécessaire de modifier le système cognitif comme dans la représentation procédurale. De plus, la même base de savoir peut être utilisée pour plusieurs buts en changeant l'interpréteur (ex: diagnostic, enseignement ou acquisition de connaissances).

Trois méthodes de représentation déclarative seront décrites ici: les règles de production, le calcul de prédicats et les réseaux sémantiques.

1.3.1 Les règles de production

La représentation d'une connaissance de type "x implique y" peut se faire par l'utilisation de règles de production de la forme:

SI "premisses" ALORS "conclusion".

Les prémisses d'une règle décrivent le contexte où la règle peut s'appliquer et la conclusion indique l'information obtenue, la décision à prendre ou l'action à poser dans ce contexte. Les règles de production contiennent tout le savoir à propos de la résolution d'un problème donné: les connaissances sur le domaine et comment les utiliser. Voici une règle de MYCIN exprimée en français courant:

Si 1/ le site de la culture est le sang,

et si 2/ l'organisme est Gram négatif,

et si 3/ l'organisme est de forme batonnet,

et si 4/ le patient est un hôte à risque,

alors , il est probable (0,6) que l'organisme est le pseudomia aeruginosa.

Les règles de production sont utilisées en relation avec les faits de l'espace de travail pour inférer de nouveaux faits. On cherche les antécédants d'une règle dans la base de faits. S'ils sont trouvés, la règle est appliquée, i.e. l'action est exécutée et modifie l'espace de travail. Le système cognitif détermine dans quel ordre appliquer les règles selon des stratégies généralement simples (ordre pré-déterminé, ordre des règles, etc....).

Cette façon de procéder permet de justifier facilement le raisonnement ayant mené à une conclusion, il suffit de chainer à rebours les règles depuis les conclusions jusqu'aux faits. Il est facile de procéder de même pour justifier une demande d'information à l'usager.

A chaque règle est associé un coefficient d'atténuation dont la valeur varie en 0 et 1. Ce coefficient mesure le degré de certitude de l'expert en la conclusion, si cette règle est appliquée. Ceci permet de calculer la vraisemblance de chaque assertion.

Nous verrons plus en détail comment fonctionne un système expert à base de règles dans la deuxième section de cet exposé.

1.3.2 Le calcul des prédicats

Une règle de production "premisses → conclusion", une façon de représenter un théorème: c'est un énoncé de logique mathématique qui peut être vrai ou faux. Dans un S.E., toutes les règles sont supposées vrai comme des axiomes. Les règles de production sont très proches des grammaires formelles de la logique mathématique.

Il serait intéressant de développer un système expert en se plaçant dans les champs de la logique mathématique où l'élément de base est la clause ou conjonction de prédicat, et, où le raisonnement se fait sous forme de démonstration de théorème. Même si plusieurs S.E. furent développés en utilisant PROLOG, un langage basé sur ce formalisme, l'intérêt demeure avant tout théorique. Il demeure en effet difficile de programmer un système dans un langage, certes très puissant, mais dont le niveau de structuration est équivalent à celui d'un assembleur.

Le système cognitif d'un tel système est un système de démonstration automatique de théorème de logique.

1.3.3 Les réseaux sémantiques

Le réseau sémantique est une méthode pour représenter les connaissances sur des relations entre entités. Il utilise des partitions comme façon de grouper significativement des parties du réseau. Cette représentation possède tout le pouvoir du calcul des prédicats qu'elle peut facilement représenter. De plus, les réseaux sémantiques possèdent d'autres propriétés intéressantes: la représentation directe des relations ensemblistes, facilité d'accès, facilité d'accès aux variables, etc.

Un réseau sémantique consiste en des noeuds reliés par des arcs dirigés. Il y a deux types de noeuds: noeud-objet (échantillon, roche, minerai, etc...) et noeud-relation (ET, OU, EST, COMPOSE DE, etc...). Les arcs entre le noeud-relation et les noeuds-objets permettent de représenter le rôle de l'objet par rapport à la relation (valeur, prémisse, conclusion,...) grâce à son étiquette. Une relation entre deux objets est représentée par trois noeuds: un noeud-relation indiquant la relation (est composé de, est au-dessus de, etc.) et deux noeuds-objets indiquant les objets en relation. Le noeud-relation est connecté aux noeuds-objets par deux arcs étiquetés du rôle des objets dans la relation.

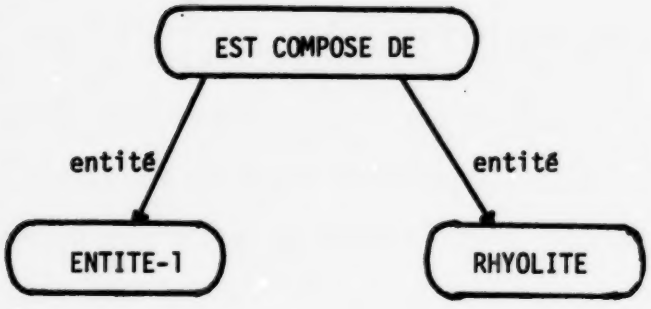

FIGURE 2

Représentation de la prémisse: "L'entité-1 est compose de rhyolite".

Certaines relations telles les relations ensemblistes: "est sous-ensemble" de et "est élément de" sont tellement fréquentes qu'elles ne sont pas réprésentées par trois noeuds et deux arcs, mais plutôt par un arc étiqueté du type de relation ($\subset$ ou $\in$) entre les deux noeuds-objets. De cette façon, les noeuds peuvent être classifiés de façon hiérarchique. Par exemple, l'ensemble suivant de relations permet de construire le réseau de la figure 3. Le rhyolite est une roche ignée. Les roches ignées sont des roches. Les roches sédimentaires sont des roches. Le marbre est une roche sédimentaire. Les minéraux sont des matériaux. Les roches sont des matériaux.

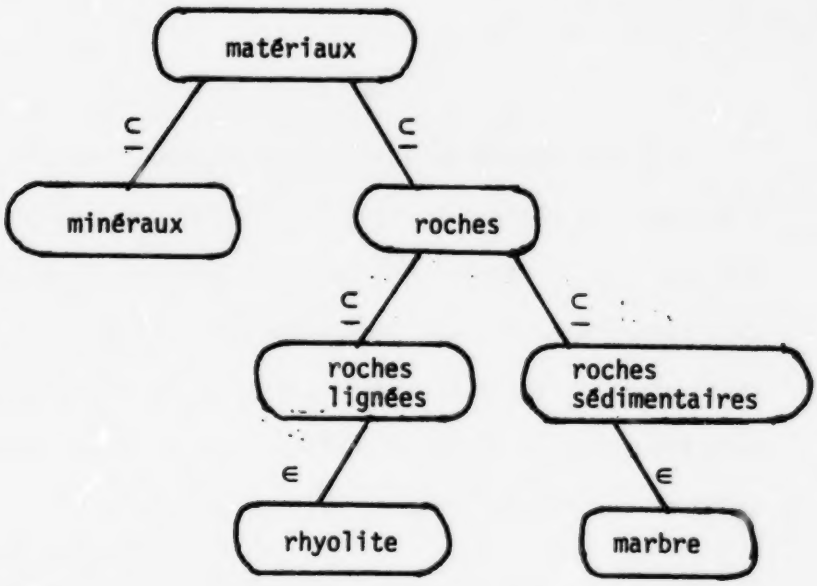

FIGURE 3

Un réseau hiérarchique de relations

Cette classification fournit une information structurelle sur les connaissances contenues dans le réseau de plus, chaque type de relation possède son descripteur qui caractérise la relation, ainsi que le rôle de ses arguments par un pointeur vers le sommet de la classe de chaque argument. Une fois construit, le réseau sémantique contient le savoir structurel et le savoir de support.

Au début du raisonnement, l'état des relations est indéterminé. Les informations recueillies permettent de déterminer la véracité d'un certain nombre de noeuds-relations. Si une relation existe entre des prémisses déterminées et une conclusion, nous pouvons inférer sur la conclusion. De plus, l'information peut être propagée par héritage le long des relations ensemblistes.

S'il est facile de voir si une hypothèse et une conclusion sont liées par une règle, il est difficile de s'assurer,dans un réseau sémantique, si la règle a été utilisée dans le raisonnement ou si les deux éléments sont vrais indépendamment. Dans un cas extrême, l'hypothèse pourrait avoir été démontrée après la conclusion. Ceci pose des problèmes lors de la justification, car le raisonnement expliqué peut différer considérablement du raisonnement utilisé.

Cette méthode de représentation a été utilisée pour PROSPECTOR [DUDA 78], un S.E. pour la recherche de gisements de minéraux.

1.4 La représentation mixte

Les modes de représentation procédurale et déclarative ont chacun leurs avantages et leurs défauts qui sont souvent complémentaires, par exemple, la représentation procédurale est fortement hiérarchisée , alors que la représentation déclarative l'est très peu. Il peut être avantageux de combiner ces deux modes de représentation pour en former un troisième mixte. Un de ces modes, le "frame ", ressemble à un réseau sémantique généralisé. Les niveaux supérieurs du réseau contiennent les objets, les concepts généraux et les niveaux inférieurs et sont constitués de cases pouvant recevoir des données sur les composantes de base. Les éléments de divers minéraux sont reliés par des liens représentant des

relations (est un, élément de, est dessus, est un sous-type de, etc.). Les propriétés sont héritées en parcourant le réseau et un élément peut avoir de multiples ancêtres et être utilisé par plusieurs concepts.

Dans un "frame", il est possible d'avoir des procédures effectuant automatiquement des transformations permettant des visions différentes d'un même élément. Si une date est entrée dans un "frame" sous forme jour-mois-an, la date peut devenir accessible sous la forme jour-an ou en lettres selon les spécifications du "frame". Le "frame" permet de représenter une situation et les connaissances par rapport à cette situation. Il permet aussi d'effectuer les transformations des données utiles dans cette situation pour compléter le raisonnement.

1.4.1 Les prototypes

Lors du développement du S.E. PUFF pour le diagnostic et le traitement des maladies pulmonaires, à partir de la méthodologie de MYCIN, plusieurs difficultés sont survenues. Dans les systèmes experts fondés sur les règles de production, il est difficile de représenter des connaissances supplémentaires, sans tenir compte de la totalité des règles ou de changer l'ordre des demandes d'information selon le contexte. Ces difficultés rendaient la performance de PUFF beaucoup moins intéressante que prévue. Selon le contexte, l'ordre logique des questions pour un

médecin diffère considérablement, PUFF ne réussissait pas à tenir compte du contexte et, de ce fait, avait un comportement illogique pour les médecins, car il posait trop de questions inutiles.

Pour permettre une gestion explicite du raisonnement selon le contexte, tout en conservant les avantages de la représentation par règles de production, Aikins [AIKI 83] a développé une nouvelle forme de représentation du savoir: le prototype qui est un "frame" simplifié. Un prototype est constitué d'une série de cases (variables, pointeurs, textes, appels de procédure, etc.) permettant la description du contexte, la représentation de savoir relatif à ce contexte. Un prototype étant un "frame", une même case peut être utilisée par plusieurs prototypes.

Les cases contiennent des textes explicatifs sur le contexte des pointeurs vers des prototypes plus spécifiques, des règles de production pour inférer, des valeurs de défaut, les cas spéciaux, des calculs de probabilité, etc. L'appel à un prototype déclenche une série connue d'actions déterminées par le contexte, le modèle et les données. Ces actions, appel de règles, de procédures ou de prototypes effectuent un travail précis (inférence, traitement, vérification du contexte, calcul de l'incertitude, etc.)

Le contrôle du raisonnement est toujours indépendant du savoir, mais le prototype permet facilement l'adaptation du raisonnement au contexte, l'utilisation de procédures pour effectuer certaines parties du raisonnement

et la définition de configurations probables des données facilitant ainsi la détection des cas spéciaux. Le modèle du prototype permet aussi un plus grande facilité dans la modification du savoir à cause de la modularité des prototypes.

Après cette description brève des systèmes experts et de leurs composantes, nous allons décrire dans la section suivante,plus en détail, le fonctionnement de systèmes experts basés sur deux modes de représentations: les règles de production et les prototypes.

SECTION 2

Plus de détails

Deux modèles de représentation du savoir présentent un intérêt particulier pour la recherche opérationnelle: les règles de production et les prototypes. Le second est une généralisation du premier. Il est possible de concevoir un prototype qui est un système de règles de production.

Les règles de productions permettent de décrire des algorithmes sous forme Si " prémisse" ALORS "action". Si,l'exemple suivant pour le calcul des chemins les plus courts, est particulièrement simple et inefficace, il n'en est pas toujours de même.

Règle 1: Si nous voulons calculer le chemin le plus court entre i et j.

Alors effectuer dans cet ordre,

1) initialiser $\Pi_{ij} = 0$ et $\forall \neq K$ i $\Pi_K = \infty$

2) appliquer les autres règles.

Règle 2: Si $i = j$

alors Π_i est la réponse cherchée.

Règle 3: Si $\forall(k,l) \in A$ tel que $\pi_k + C_{kl} \geq \pi_l$
alors π_j est la réponse cherchée.

Règle 4: Si $(k,l) \in A$ tel que $\pi_k + C_{kl} < \pi_l$
alors $\pi_l := \pi_k + C_{kl}$

Un algorithme bien connu permet d'obtenir le même résultat très rapidement. Toutefois, pour le contrôle d'algorithmes heuristiques, il pourrait être intéressant de définir ceux-ci sous forme déclarative.

Les prototypes permettent de traiter différemment les données selon le contexte. Ceci peut correspondre en R.O. à utiliser des algorithmes différents pour des contextes différents. Par exemple, le répartiteur d'une compagnie de transport urbain n'utilise pas le même schéma de pensée s'il tente de créer des horaires de travail pour répondre à la période de pointe ou à la charge de base. Une heuristique utilisant des prototypes pourrait créer différents patrons de solution selon le contexte.

2.1 Les règles de production

Les systèmes experts utilisant les règles de production seront maintenant examinés plus en détail. Ce modèle de représentation du savoir utilise des implications "x implique y" pour représenter les connaissances. Ces implications sont représentées par des règles de production de de la forme: SI "prémisses" ALORS "conclusion".

Les prémisses permettent de décrire le contexte dans lequel la conclusion s'appliquera. Chaque prémisse est un quadruplet "prédicat" "objet""attribut""valeur". Le prédicat est une relation entre l'attribut de l'objet et la valeur par exemple, cette relation peut être: est, est le même que, appartient à, etc. L'objet est celui sur lequel porte l'information par exemple, dans un contexte médical, l'objet peut être un organisme, un médicament, une culture, le patient, etc. L'attribut est une propriété de l'objet sur laquelle nous avons de l'information; toujours dans un contexte médical, ce peut être l'identité, la sensibilité, le site, etc. La valeur est l'information connue par rapport aux trois composantes précédentes. Voici des exemples de prémisses ou de conclusions exprimées en langue courante:

- Le site de la culture est le sang.
- L'organisme est de forme bâtonnet.
- Le site de la culture était stérile.
- La thérapie recommandée doit être choisie parmi la tétracycline, la lindomyrine ou la gentamicine.

Une règle est composée d'une série de prémisses et d'une conclusion. A chaque règle est associé un coefficient d'atténuation, dont la valeur varie entre 0 et 1. Ce coefficient mesure le degré de certitude de l'expert en la conclusion, étant donné les prémisses. Ces coefficients permettent de calculer la vraisemblance de la conclusion selon une méthode qui sera vue à la troisième section de cet exposé.

L'interprète des règles peut effectuer sa recherche de conclusion selon deux schèmes de raisonnement:

1) par recherche en avant. L'interprète fouille parmi les règles jusqu'à ce qu'il trouve une règle dont les prémisses sont vérifiées par les connaissances contenues dans l'espace de travail. Si l'interprète ne peut trouver une telle règle, il détermine l'information supplémentaire permettant de tirer le plus de nouvelles conclusions et demande cette information à l'usager.

2) par recherche en arrière. L'interprète choisit un objectif et fouille parmi les règles pour trouver celles dont les conclusions peuvent impliquer cet objectif. Si les prémisses d'une de ces règles correspondent aux connaissances contenues dans l'espace de travail, alors cet objectif est atteint et cette partie de raisonnement est terminée. Sinon, la prémisse manquante devient un sous-objectif et la procédure est appliquée récursivement. S'il est impossible de résoudre un objectif, alors le système cognitif demande une information supplémentaire à l'usager.

Pour raisonner, l'interprète dispose de connaissances sur sa représentation du savoir. A chaque objet est associé un descripteur énumérant les attributs possibles pour cet objet, tous les attributs n'étant pas possibles pour tous les objets. Il est impossible de parler de sexe de la culture, cet attribut se rapportant au patient.

De même à chaque prédicat est associé un profil permettant au système de savoir quel couple attribut-objet commande le prédicat en question. Les profils permettent une évaluation rapide en indiquant si la valeur d'un couple est connue ou non. Puisque chaque couple se retrouve dans plusieurs prémisses, dans plusieurs règles, les profils permettent, par une logique simple, d'indiquer si une règle est applicable, en attente ou inappropriée dans la situation courante.

Un système expert construit selon ce modèle de base pourrait être considéré comme un analogue des ordinateurs des premières générations. En effet, ce S.E. manque de structure explicite. Toutes les règles n'ont pas le même rôle, certaines sont utilisées pour diriger le raisonnement, d'autres contiennent des connaissances. Voyons maintenant les principales critiques et les solutions apportées face à ce système expert de base.

Le savoir, présenté sous forme de règle de production, n'utilise pas la structuration présente dans l'esprit de l'expert consulté. L'expert peut, en effet, diviser ses connaissances selon plusieurs niveaux [HAYE 79, WILE 81, STEF 81b, CLAN 83]. De façon générale, le premier niveau concerne les connaissances stratégiques et vise à ordonner les objectifs et les hypothèses lors de la résolution. Ce niveau est souvent indépendant du problème traité, mais dépend plutôt du type d'application (diagnostic et traitement, design, etc...). Ce niveau de connaissance est

appelé méta-savoir et voici un exemple de méta-règle exprimé en langue courante:

"Si l'on recherche une thérapie
alors, dans cet ordre, considérer les règles qui
permettent de:

1. acquérir de l'information clinique sur le patient;
2. trouver quels organismes, s'il en existe, sont cause de l'infection;
3. identifier les organismes les plus vraisemblables;
4. trouver tous les médicaments potentiellement utiles;
5. choisir les plus acceptés en plus petit nombre."

Le méta-savoir est constitué des connaissances à propos de la résolution de problèmes qui peuvent être énoncés,indépendamment du domaine, en terme de méta-objectifs et de méta-plans. Le méta-savoir permet d'examiner une condition, d'y détecter des fautes ou des améliorations possibles et de proposer un nouveau méta-objectif et les méta-plans associés pour changer la conclusion. Un exemple de thème est le cas où plus d'un trai- est disponible, le méta-objectif déclenché est de choisir le plus désirable et le méta-plan associé consiste à évaluer la désirabilité et à choisir.

Le savoir sur les connaissances relatives au domaine peut lui aussi se diviser en niveaux. De façon générale, deux niveaux sont distingués: le savoir structurel et le savoir de support. Le premier niveau, le

savoir structurel permet d'ordonner les connaissances du domaine. Il s'agit de connaissances sur les relations entre les attributs, les objets et les prédicats, les descripteurs et les profils font partie du savoir structurel. Le savoir structurel comprend aussi des règles pour guider la recherche. Ces règles listent l'ordre habituel de recherche (les causes communes avant les inhabituelles), les exceptions et de façon générale les informations permettant de mettre à profit le méta-savoir.

Le dernier niveau du savoir, le savoir de support contient les connaissances dépendant du domaine et donnant une information de bas niveau par exemple, un rapport causal entre un symptôme et un diagnostic particulier. Les règles décrivent des propriétés de classes, des connaissances générales et spécialisées au domaine et les jugements basés sur des probabilités, des rapports coût/bénéfices et l'expérience de l'expert.

Ce niveau de savoir peut se diviser en sous-niveaux dépendant du domaine d'application par exemple, HEARSAY-II [HAYE 79], un S.E. permettant la reconnaissance de la parole divise ses connaissances tout naturellement en sons, syllabes, mots et phrases. D'autres domaines d'applications auront les sous-niveaux propres à une bonne gestion du savoir.

Cette division des connaissances en niveaux qui sont définis chacun par rapport à son niveau inférieur permet une simplification de la manipulation des connaissances. Une analogie existe à ce propos avec les

systèmes informatiques où il existe un langage machine, un langage assembleur, des langages de programmation et un langage de contrôle pour nous permettre de simplifier notre utilisation du système informatique.

L'autre grand problème des S.E. avec règles de production est de connaître l'ordre d'exécution des tâches. Dans une procédure, la prochaine instruction à exécuter est parfaitement définie. Dans un S.E., il est souvent difficile de déterminer la prochaine tâche de l'agenda qui sera exécutée. L'interprète détermine la prochaine tâche selon un ordre implicite comme dans une boîte noire. Cet ordre peut être déterminé par un usager en plaçant les prémisses dans les règles ainsi que les règles dans le bon ordre. Cette façon de procéder est très complexe et pose de nombreux problèmes d'ordonnancement.

Il est possible de retirer cette responsabilité à l'interprète en établissant un système de priorités pour chaque tâche. Ces priorités sont modifiées pour refléter les conditions changeantes de l'état du S.E. L'établissement de ces priorités relève du méta-savoir et structurel.

Un bon exemple de la division de la base de savoir pour simplifier la complexité apparente des connaissances est le S.E. MOLGEN [STEF 81A, STEF 81B] qui planifie des expériences de synthèse en génétique moléculaire. Pour planifier une expérience, le système doit déterminer les appareils et les substances à utiliser, tout en déterminant l'ordre des manipulations à effectuer. La planification doit tenir compte de la disponibilité

des appareils et de la comptabilité des substances utilisées. Le design s'effectue du plus général au plus détaillé par raffinement successifs.

Le système est divisé en 4 niveaux fonctionnant chacun avec son système cognitif, son espace de travail et ses connaissances. Nous allons examiner ces niveaux du plus général au plus détaillé. Certains de ces messages posent des questions au niveau inférieur, les autres indiquent les choix du niveau supérieur.

1) Niveau de la stratégie

Ce niveau est très indépendant du domaine. Il comprend deux cycles de contrôle, le premier d'engagement minimum et le second de choix heuristique. Il existe 4 opérateurs à ce niveau: DEVINER et DEFAIRE,dans le cycle heuristique, TROUVER et REDEMARRER, dans le cycle de moindre engagement.

TROUVER est l'opérateur qui trouve des tâches à accomplir en demandant à tous les opérateurs du niveau inférieur, celui de planification, d'en proposer. Le choix de l'opérateur de planification à déclencher est effectué grâce à des priorités sur ces opérateurs. Une tâche non-choisie est suspendue. Si le plan devient irréalisable, un choix arbitraire (dans DEVINER) était mauvais et il faut retourner en arrière (DEFAIRE) et faire un autre choix. TROUVER ne prend que des décisions réalisables.

REDEMARRER repart les opérateurs suspendus et termine le travail inachevé. Ces tâches avaient été suspendues, car conformément à la stratégie de moindre engagement, nous avons préféré attendre avant de prendre une décision prévoyant recevoir plus d'informations à mesure que la planification se précisait.

DEVINER est utilisé lorsque durant la planification il n'y a plus de décision justifiable. Nous devons alors choisir arbitrairement Cet opération procède comme TROUVER et choisit une des tâches possibles.

Le contrôle sur ce niveau est effectué grâce à un automate fini non-déterministe à quatre états, les quatre opérateurs. Tant que cela est possible, nous restons dans la partie déterministe (TROUVER et REDEMARRER). S'il est impossible de faire un choix déterministe, nous allons à l'état DEVINER. Si le plan devient irréalisable, nous allons à l'état DEFAIRE.

2) Niveau de planification

Ce niveau contient le savoir sur le design du plan d'expérience. Ce niveau est contrôlé par le niveau de stratégie et contrôle le niveau du domaine abstrait. Ce niveau opère sur des opérateurs de planification, i.e. pour créer et arranger les étapes dans l'espace du domaine. Dans MOLGEN, la planification peut être vue comme une série d'opérations sur les contraintes. La FORMULATION des contraintes exprime une description de l'espace de solution désiré. Ceci corres-

pond à une décision dans la planification. Le système crée une description partielle de la contrainte et remet la décision finale à plus tard. La PROPAGATION des contraintes communique aux différentes parties du plan (sous-problème) les décisions prises. Ceci permet d'indiquer à un sous-problème qu'avant de raffiner son plan, il doit tenir compte d'un autre sous-problème. La SATISFACTION de la contrainte détaille les entités abstraites en objets ou opérateurs spécifiques.

Les opérateurs de planification sont divisés en trois classes. La première classe compare les objectifs avec les résultats obtenus en simulant le plan actuel de façon à trouver des différences. Ceci permet de déterminer de nouveaux objectifs. La seconde classe des opérateurs vise à déterminer comment remplir les nouveaux objectifs proposés par l'application de la première classe. Ces opérateurs visent donc à choisir une tâche à effectuer dans le domaine abstrait, à spécifier les sous-objectifs de cette tâche et à vérifier par simulation que la tâche choisie peut remplir les objectifs. S'il est impossible de choisir une tâche, par manque d'information, le contrôle est retourné au niveau de stratégie et le choix suspendu. La troisième classe d'opérateurs de planification contrôle le raffinement du plan en imposant, propageant et vérifiant la satisfaction des contraintes. Ces trois classes correspondent à trois étapes: trouver les objectifs, tester et choisir différentes façons de remplir ces objectifs et finalement appliquer la décision prise.

3) Niveau du domaine abstrait

Ce niveau contient le savoir sur les objets et sur les opérations possibles sur ces objets. Les opérateurs sont divisés en quatre classes abstraites utilisées par la planification. Les objets sont divisés en 84 classes (ex: antibiotiques, enzymes, gènes, etc.). A ce niveau, l'ordre des opérations-abstraites sur des classes d'objets est déjà déterminé (au moins à l'état de projet) et il ne reste qu'à raffiner le plan, tout en respectant les contraintes.

Le plan est détaillé en choisissant d'abord l'opérateur dans la classe. Ceci peut introduire une contrainte sur le choix des objets dans leur classe. Lors de la propagation de la contrainte sont éliminées des classes d'objets concernés, les objets ne respectant pas la contrainte, si une classe devient vide, le plan est irréalisable. Lors de la satisfaction de la contrainte, un choix définitif de l'objet est effectué parmi ceux qui sont encore disponibles.

4) Niveau du domaine

Ce niveau est contrôlé par le ninveau du domaine abstrait et contient tous les objets et opérations.

La division du savoir en niveaux permet de définir la solution progressivement, en vérifiant par simulation l'exactitude des choix et la disponibilité des objets.

2.2 Les prototypes

Les prototypes ont été créés pour mettre une gestion des raisonnements par les règles variant selon le contexte. Le mode de contrôle explicite conserve les avantages de justification facile du raisonnement.

Un prototype est composé d'une série de cases et bornes spécifiant les composantes attendues. Ces cases fournissent un emplacement explicite pour l'information, ceci permet de déterminer facilement l'information manquante et les bornes permettent de déterminer si la valeur spécifiée pour la case est normale dans le contexte du prototype. De plus, le système peut juger plus facilement quand il a suffisamment d'informations pour compléter son raisonnement.

Le but global du S.E. est de faire correspondre les caractéristiques du problème actuel avec un ou plusieurs des prototypes. Le but à court terme du S.E. par rapport à l'information la plus importante à obtenir est déterminé par le savoir de contrôle et les valeurs des cases contenues dans le prototype présentement exploré. Le prototype contient une série, déterminée par le savoir structurel, de cases contenant ou destinées à recevoir diverses informations: des informations sur le patient, des règles ou des procédures pour inférer de nouveaux faits, des informations sur le contrôle du raisonnement dans le prototype, etc.

L'appel à un prototype déclenche une série prédéterminée par le prototype, le savoir et les données, d'actions: appels de règles, de procédures ou de prototypes. Ces actions effectuent un travail précis d'inférence, de transformation de données, de vérification du contexte, de calcul des coefficients de vraisemblance, etc. Chacune de ces tâches est prises en charge par une partie du prototype. Voyons maintenant comment , dans le cadre de CENTAUR [AIKI 83] un prototype est structuré et quel est le contenu des cases. Le contrôle à l'intérieur d'un prototype sera ensuite

A) les cases d'information générale. Ces cases contiennent des informations sur le prototype et sur son rôle dans le réseau de prototype. Certaines informations concernant l'histoire du prototype (auteur, date de création, source, etc.). D'autres contiennent des pointeurs vers des prototypes plus spécifiques, si nous désirons raffiner le diagnostic et des pointeurs vers des prototypes plus généraux, si le prototype activé s'avère une voie sans issue. Les dernières cases contiennent des textes permettant d'expliquer à l'usager l'hypothèse étudiée et le processus poursuivi par ce prototype;

B) les cases de contrôle contiennent les plans à exécuter pour poursuivre le raisonnement selon les différentes situations. La première case contient un plan de départ pour vérifier l'hypothèse

du prototype. Selon la conclusion obtenue, deux plans existent selon que l'hypothèse est confirmée, il est alors nécessaire d'effectuer un traitement, de poursuivre le diagnostic en allant au plus particulier. Si l'hypothèse est infirmée, le plan doit effectuer le retour en arrière en utilisant au maximum l'information obtenue;

C) les cases de règles contiennent les systèmes de règles de production utilisées pour raisonner dans différentes situations. Jusqu'à cinq systèmes de règles peuvent exister:

1) les règles de perfectionnement pour contrôler la demande des informations supplémentaires pour raffiner le diagnostic;

2) les règles d'inférence utilisent les données pour déduire de nouveaux faits;

3) les règles de déclenchement utilisent les données, les conclusions obtenues pour suggérer des prototypes à déclencher pour poursuivre la recherche;

4) les règles sur les faits non-utilisés tentent d'expliquer les faits non-utilisés dans le raisonnement et les faits qui ne correspondent pas à l'hypothèse;

5) les règles de conclusion font le sommaire des informations sur le prototype et "écrivent" le rapport de consultation.

Chacun de ces systèmes des règles peut être utilisé dans d'autres prototypes pour la même fonction.

D) les cases de composantes contiennent l'espace pour les données à obtenir. Puisque nous connaissons les données obtenues, nous pouvons déterminer les valeurs manquantes. Une des caractéristiques des prototypes, et celle qui permet de définir le contexte, est de pouvoir associer à chaque variable des bornes sur les valeurs plausibles, suprenantes ou possiblement erronées. De plus, il est possible de définir des valeurs de défaut pour ce contexte permettant d'inférer avec des variables non-déterminées;

E) la case du coefficient de vraisemblance du prototype;

F) les cases d'historique du raisonnement contiennent les détails du déclenchement du prototype. Soit le nom du prototype appelant, les raisons du déclenchement et les paramètres passés. Ceci permet de justifier facilement le raisonnement utilisé.

Le contrôle avec les prototypes dans CENTAUR est très simple. L'interprète du système exécute la tâche située au sommet de la pile de l'agenda. Le processus se répète lorsque l'exécution de la tâche est terminée. Les tâches peuvent être ajoutées à la pile de deux sources: les cases de contrôle d'un prototype et l'exécution d'une tâche peut causer l'ajout d'une autre tâche sur l'agenda. Une tâche dans l'agenda comporte la tâche à accomplir ainsi que la source de la tâche et la raison de l'exécution de la tâche.

L'exécution d'un prototype comporte près de dix étapes:

1) entrée des données initiales;

2) création de la liste des hypothèses en interrogeant les prochains prototypes pouvant être déclenchés et en calculant leurs coefficients de vraisemblance;

3) à partir de la liste des hypothèses, l'interprète choisit la tâche à ajouter à l'agenda en utilisant une des trois stratégies suivantes: stratégie de confirmation, la plus plausible d'abord, stratégie d'élimination, la moins plausible d'abord ou un ordre fixé à priori;

4) déclenchement du prototype choisi, exécution de la case de départ et des règles d'inférences pour obtenir des valeurs dans toutes les cases des composantes, que ce soit en faisant des déductions ou en posant des questions à l'usager. Si à la suite de cette étape, le déclenchement de nouveaux prototypes est suggéré pour remplir les cases de composantes, retourner à l'étape 2;

5) vérifier l'hypothèse. L'interprète vérifie la concordance des valeurs des composantes avec les valeurs plausibles du prototype. Selon le sort réservé à l'hypothèse, la case hypothèse-confirmée ou la case hypothèse-infirmée est ajoutée à l'agenda. Si ces tâches suggèrent le déclenchement de nouveaux prototypes pour spécialiser l'hypothèse ou retourner en arrière, il faut exécuter l'étape 3;

6) vérifier l'utilisation complète des faits. S'il ne reste plus de prototypes à explorer, marquer tous les faits utilisés. S'il reste des faits inutilisés, on applique les règles sur les faits inutilisés pour tenter d'expliquer ces faits;

7) amélioration du résultat. Les règles de perfectionnement sont appliquées sur le résultat temporaire pour tenter de le raffiner et d'obtenir un résultat final;

8) faire le sommaire, les règles de conclusion associées aux prototypes confirmés pour obtenir un rapport d'exécution. Ce rapport comprend les conclusions obtenues et leur justification.

Le déclenchement d'un prototype ressemble à un appel de procédure, car une série déterminée d'actions est exécutée, mais au contraire d'une procédure, les actions que démarre un prototype ne sont pas nécessairement de type procédural. Ce sont en effet souvent des systèmes de règles de production. Les prototypes permettent de modulariser les règles de production augmentant ainsi considérablement leur puissance de raisonnement et leur facilité d'utilisation.

SECTION 3

Les techniques de la recherche opérationnelle et des systèmes experts

L'utilisation des techniques traditionnelles de la recherche opérationnelle dans les systèmes experts semble peu développée. Même dans un système dont le but est l'optimisation des routes [ELLI 82] dans de grands réseaux routiers, l'emphase n'est pas mise sur l'utilisation des algorithmes de calcul de plus courts chemins qui sont jugés non-applicable en raison de la taille des réseaux. L'emphase est plutôt mise sur une utilisation judicieuse des techniques de base de l'algorithmique comme la recherche en largeur, la recherche en profondeur et diviser-pour-régner. Ceci peut s'expliquer par le caractère heuristique des systèmes experts. L'optimisation semble avoir un caractère local caractérisé par l'utilisation de critères d'économies locales.

3.1 Une méthode d'inférence

Une technique semble toutefois utilisée plus particulièrement dans les systèmes à base de règles de production: l'inférence des probabilités. L'inférence est utilisée pour tenir compte de l'incertitude entourant les probabilités sur les conclusions.

La probabilité associée à une conclusion (prémisse) est appelée coefficient de vraisemblance. L'inférence se faisant en série sur les règles Shortliffe [SHOR 76] a défini un mécanisme d'inférence non bayesien pour MYCIN.

Nous avons vu plutôt qu'à chaque règle est associé un coefficient d'atténuation (C.A.) variant entre 0 et 1 qui reflète la confiance de l'expert envers la règle. De plus, les faits concernant le problème "objet""attribut""valeur" sont valués par le coefficient de vraisemblance dont la valeur est comprise entre -1 (faux) et 1 (vrai). La valeur 0 représentant l'ignorance par rapport à ce fait.

Le C.V. d'un nouveau fait,déduit par l'application d'une règle dont le coefficient d'atténuation est C.A. et le coefficient de vraisemblance de la Ième prémisse, est CP_i est donné par:

$$C.V. = C.A. \times \min \{CP_i\}.$$

Ce modèle permet la coexistence de plusieurs valeurs différentes pour un même couple "attribut""objet". De plus, tout fait dont la valeur absolue du C.V. est inférieure à 0,2 est considéré comme peu crédible et est éliminé de la base des faits.

Si deux règles différentes, lors du raisonnement, donnent la même conclusion avec des C.V. C1 et C2, le coefficient de vraisemblance de l'union est donné par:

$$C.V. = C1 + C2 - C1 \times C2.$$

Cette méthode d'inférence n'est pas la seule utilisée, DUDA [DUDA 78] a développé un système bayesien pour PROSPECTOR.

Toutefois, l'avantage d'un système non-bayesien est de permettre de considérer simultanément plusieurs voies crédibles, i.e. la somme des coefficients de vraisemblance ou "probabilité" peut être plus grande que 1 ou plus petite que 0. Toutes les déductions se faisant entre trois états (vrai, faux ou ignorance), le rejet d'une conclusion n'implique pas qu'elle est fausse, mais plutôt qu'elle est ou fausse, ou indéterminée. Un tel système permet de déduire les questions à poser à l'usager pour augmenter notre connaissance à propos des conclusions, i.e. augmenter la valeur absolue de leurs coefficients de vraisemblance.

3.2 ALICE, un système expert

ALICE [LAUR 78] est un système expert développé pour résoudre des problèmes combinatoires, i.e. des problèmes de recherche d'un point dans un espace discret, fini, sujet à une série de contraintes. Plusieurs problèmes connus font partie de cette famille: commis-voyageur, routing, scheduling, coloration d'un graphe, sac alpin, etc. Beaucoup de ces problèmes font aussi partie du groupe des problèmes NP complet et il

n'existe pas d'algorithmes efficaces pour les résoudre. Souvent ces problèmes sont résolus en mettant au point une méthode de branchement (Branch and Bound) particulière pour chaque problème. ALICE tente de résoudre par une méthode d'intelligence artificielle toute cette famille de problèmes.

L'usager doit décrire son problème dans un langage descriptif demandant une bonne connaissance du problème à résoudre et du langage de description. Ce langage utilise des notations ensemblistes et algébriques pour décrire le problème. Ce détail rend l'usage de ALICE un peu rébarbatif. De plus, il manque à ALICE la capacité d'expliquer son raisonnement.

Généralement, un problème combinatoire est exprimé sous forme d'un graphe, sous forme de formules algébriques ou sous forme de tests sur espace discret. ALICE utilise ces trois représentations simultanément. Le problème est représenté comme une fonction $y = f(x)$ assujettie à une série de contraintes où y appartient à A, l'ensemble solution et x appartient à D, le domaine.

Lors de la compilation du problème, le système crée un graphe bipartite de D à A où les arcs représentent des valeurs possibles de la fonction. Tous les acrs représentant des valeurs impossibles sont éliminés. D'autres contraintes sont formulées sous forme de formules algébriques simples. Les contraintes les plus complexes sont conservées sous forme de tests.

La valeur complète pour tous les points de la fonction est déterminée par un processus d'essai-erreur. A chaque étape, une valeur de la fonction pour un point est partiellement ou complètement déterminée selon l'une des stratégies disponibles. ALICE vérifie alors l'effet sur le problème de cette décision. Des arcs peuvent devenir impossibles, d'autres sont déterminés, le problème peut devenir irréalisable obligeant à retourner en arrière. ALICE doit aussi vérifier l'effet sur les formules algébriques qui peuvent être simplifiées, pouvant même se transformer en contrainte sur un arc et être incorporées au graphe. Les tests peuvent aussi être simplifiés pour devenir une formule algébrique. La formulation du problème évolue dynamiquement lors de la résolution permettant d'utiliser la formulation la plus simple possible à l'étape poursuivie.

ALICE permet d'obtenir une solution réalisable et de l'améliorer jusqu'à l'obtention d'une solution optimale dans la plupart des cas. ALICE a permis la résolution de plus d'une centaine de problèmes combinatoires dont la complexité allait du problème de mathématique amusant à de grands problèmes de scheduling.

3.3 L'utilisation de la R.O.

L'utilisation des techniques classiques de la R.O. dans les systèmes experts est négligeable et est confinée à l'utilisation de critères d'optimisation locaux dans les heuristiques. Il est significatif à ce propos

qu'à l'université Stanford, de nombreux systèmes experts (MYCIN, PUFF, PROSPECTOR, CENTAUR) aient été développés dans le cadre du "Stanford Heuristic Project".

Même si dans les sytèmes experts, on rencontre plusieurs réseaux: réseaux de règles, arbre de recherche, il n'est presque jamais question d'autres techniques que de la recherche en largeur ou en profondeur. Les critères utilisés pour le choix sont toujours locaux comme dans les algorithmes gloutons. Il serait probablement avantageux d'utiliser des méthodes garantissant de meilleures performances, telle la programmation dynamique pour obtenir de bonnes approximations. La programmation dynamique permettant d'organiser les recherches en largeur et en profondeur est une façon de vérifier l'optimalité. Cette solution pourrait être retenue pour vérifier l'impact de connaître une information sur un réseau de règles. Cette stratégie pourrait permettre de poser un nombre minimum de questions à l'usager.

Même ALICE [LAUR 78] qui résout des problèmes d'optimisation combinatoires n'utilise pas de procédure d'optimisation autre que le calcul de coûts réduits. Pourtant ALICE, qui résout ses problèmes en construisant une fonction quasi-couplage par séparation et élimination, pourrait s'inspirer des algorithmes de couplages pour construire des solutions à l'aide de chaînes alternées augmentantes. Cette façon de procéder permettrait probablement d'obtenir plus rapidement des améliorations à la solution réalisable.

Généralement, le défaut principal attribué aux techniques classiques de la recherche opérationnelle est de demander trop de conditions théoriques sur les données par exemple, la convexité pour garantir les résultats. Même en l'absence de ces conditions, les techniques peuvent être utilisées pour obtenir une solution sans garantie d'optimalité. Ces méthodes seront alors toujours supérieures à celles utilisant des critères locaux.

Pour trouver d'autres applications possible de la R.O. aux systèmes experts, il suffira d'examiner leur fonctionnement pour que toute une génération de nouvelles applications de la R.O. surgissent dans ce domaine jusqu'ici resté vierge.

SECTION 4

Le problème de routing: un exemple d'application possible

Le problème général de routing est facile à énoncer. L'usager dispose d'un ensemble de noeuds et d'arcs qui doivent être desservis par une flotte de véhicules. Le problème est de construire, pour chaque véhicule, une route réalisable, tout en minimisant le coût total. Une route est une suite d'éléments (noeuds et arcs) que le véhicule doit desservir tout en respectant diverses contraintes.

L'énoncé reste vague sur tous les points sauf un: optimiser les fonctions de coût. Un grand nombre de cas particuliers du problème de routing peuvent être définis. Les véhicules, selon le type de service à rendre, doivent parcourir des arcs, ex: ramassage des détritus, visiter des noeuds; ex: livrer des marchandises à un client, ou effectuer une combinaison des deux. Les contraintes peuvent être de divers types. Elles peuvent porter sur l'ensemble de la solution, ex: borne sur le nombre de véhicules; ou limiter chaque route, ex: capacité maximale de chargement d'un véhicule ou longueur maximale d'une route. Finalement, des contraintes peuvent être associées à chaque noeud ou arc telles des contraintes de temps (un client doit être visité avant une heure donnée) ou de précédence (un client doit être visité

avant un autre). D'autres cas particuliers existent, une revue de littérature extensive est disponible [BODI 83]; Bodin et Al. y décrivent des méthodes de résolutions exactes et heuristiques pour la plupart de ces problèmes.

Tous ces problèmes sont voisins et les méthodes de solutions particulières mises au point pour les résoudre suivent deux grands courants. La première tendance permet d'obtenir des solutions optimales, en formulant mathématiquement le problème et en résolvant cette formulation par diverses techniques de la programmation mathématique. Les techniques utilisées incluent la programmation linéaire en nombres entiers, l'énumération implicite, divers algorithmes d'optimisation sur des réseaux et la relaxation lagrangienne. L'avantage de ces approches est de permettre de trouver une solution optimisant la fonction des coûts et de garantir l'optimalité de la solution obtenue. Toutefois, ces méthodes présentent un grand désavantage, elles ne sont pas flexibles et ne permettent pas de considérer facilement des modifications très simples du problème. De plus, le temps de résolution augmente généralement très vite avec la taille des problèmes.

La seconde tendance, utilisée surtout dans les heuristiques, construit des assemblages de diverses procédures élémentaires pour résoudre les problèmes. Les procédures employées peuvent être des algorithmes de base bien documentés tels les chemins les plus courts, l'arbre de coût minimum, l'affectation, le sac alpin, etc.ou bien des concepts simples d'économie à réaliser

en ajoutant un client à une route, d'échanges de clients, de regroupement de clients en vue de former une route, etc. Généralement, la méthode mise au point trouve une solution par une suite de décisions prises en examinant l'état actuel de la solution (coût, état des contraintes, etc.) et les conséquences possibles des décisions sur la solution finale. De telles méthodes permettent généralement d'obtenir de bonnes solutions et ont généralement l'avantage d'accepter facilement des petites contraintes supplémentaires. Toutefois, il n'existe pas de logiciel général pour résoudre ces problèmes et chaque nouveau problème à traiter implique la fabrication d'un nouvel assemblage, car l'utilisation d'un tel logiciel, hors du contexte où il a été développé et évalué, produit souvent des résultats de piètre qualité.

Si les systèmes experts ont été développés pour échapper à l'attitude "boîte noire" de la première tendance, la seconde tendance se prête bien à une implantation par S.E. Les connaissances de l'expert nous permettront de construire et d'expliquer la meilleure méthode heuristique possible pour résoudre un problème particulier en choisissant les concepts ou algorithmes utiles et en organisant le passage des données entre les composantes choisies. De plus, l'interaction avec l'usager permet de créer une solution répondant aux conditions supplémentaires, par exemple, ne pas servir certains clients par le même véhicule ou ne plus utiliser un bout de route présentement en réparation, etc.

Le système expert proposé effectuera le design de routes économiques tout en respectant les critères fixés par l'usager. Le but de ce système se rapproche de celui de MOLGEN [STEF 81a, STEF 81B], un S.E. effectuant le design d'expériences de génétique moléculaire. L'approche décrite à la section 2.1 pourra être utilisée pour créer la solution de problèmes de routine.

4.1 Le design du système expert

Le choix du modèle de représentation des connaissances est primordial lors de la conception d'un système expert. Cette décision détermine presque totalement le mode de fonctionnement des autres composantes du S.E.: espace de travail, système justificateur et système cognitif. Il est important de choisir un modèle de représentation des connaissances s'harmonisant bien avec le domaine d'application.

Les connaissances dont nous disposons comprennent des algorithmes élémentaires, des concepts et un corpus d'heuristiques pour résoudre des problèmes particuliers de routing. Ce savoir semble être de type procédural. Il semblerait contre-indiqué d'utiliser, sans la modifier, l'approche de MOLGEN utilisant des règles de production divisées en plusieurs niveaux de connaissances (stratégie, planification, domaine abstrait, domaine). Il semble aussi contre-indiqué d'utiliser un savoir uniquement procédural à cause des inconvénients, exposés à la section 1.2, d'une telle approche. Le choix portera donc sur un modèle de représentation mixte: les prototypes.

Les prototypes [AIKI 83], décrits à la section 2.2, permettent l'utilisation de règles de production pour déterminer l'approche à suivre et celle des procédures pour traiter les données. De plus, ils permettent de diviser le savoir en couches, de hiérarchiser les prototypes, comme dans MOLGEN.

La division du savoir en plusieurs couches permet de réduire la complexité apparente du problème si chacune des couches correspond à une phase logique du problème pour l'usager. Dans le cas du routing, les couches suivantes semblent appropriées: la stratégie, la planification et la fabrication des routes.

Chaque niveau du savoir fonctionne comme un système agissant en interrelation avec les autres niveaux. Un niveau donné accomplit une tâche précise, commandée par le niveau supérieur, en effectuant un choix parmi les suggestions faites par le niveau inférieur. A chacune des suggestions est associé un coefficient d'efficacité mesurant la "probabilité" que la méthode suggérée produise une bonne solution pour la tâche commandée. Le système cognitif effectue son choix selon une politique de moindre engagement, remettant à plus tard toutes les décisions non-urgentes de façon à disposer du maximum d'informations lors de la décision. Cette politique permet de restreindre le nombre de choix arbitraire et ainsi de minimiser le nombre de retour en arrière en cas de mauvais choix. Plus d'une suggestion peut être retenue par le système cognitif, chacune des méthodes suggérées pouvant offrir des avantages particuliers par exemple: des routes

économiques versus des routes dont la charge est équilibrée. Le choix définitif est alors remis à plus tard lorsque les solutions seront connues et qu'il sera possible d'effectuer un bon choix avec plus de certitude.

4.2 Quelques éléments du savoir

Le niveau de la stratégie est indépendant du problème et tente à l'aide des suggestions de la planification de déterminer une stratégie générale pour résoudre le problème. Une méta-stratégie possible pourrait être composée des étapes suivantes:

1) ENONCER le problème, i.e. connaître ses principales caractéristiques, i.e. l'objectif (visiter un ensemble de noeuds ou d'arcs), la fonction objective, les contraintes présentes (charge maximale des véhicules, types de véhicules, visite à un sous-ensemble de noeuds, longueur des routes, nombre maximum de véhicules, etc...). Cette étape permet de déterminer les objectifs à atteindre.

2) DETERMINER les caractéristiques importantes. Toutes les contraintes n'ont pas la même importance. Il s'agit ici de trouver les contraintes les plus restrictives.

3) TROUVER une méthode de résolution et l'appliquer. Le niveau de planification suggère des méthodes pouvant résoudre un problème composé des caractéristiques importantes du problème original.

4) VERIFIER que la solution obtenue correspond aux objectifs. Nous vérifions que la solution du problème réduit est une solution pour le problème original. Si oui, soumettre la solution à l'usager.

5) EST-CE QUE des modifications mineures peuvent rendre la solution obtenue conforme aux objectifs? Le niveau de planification suggère des modifications pour rencontrer les objectifs manquants. Si oui, effectuer ces modifications et soumettre la solution à l'usager.

6) TROUVER une nouvelle caractéristique importante et retourner à l'étape 3. A l'étape 2, une caractéristique avait été ignorée, il faut déterminer un nouvel ordre d'importance des contraintes et essayer de trouver une nouvelle méthode de résolution.

Cette méta-stratégie est appliquée jusqu'à ce que l'usager soit satisfait de la solution. Il faut noter que le processus énoncé ici est indépendant du domaine. Il n'y a que les notes explicatives qui dépendent du problème.

Les connaissances du niveau de planification doivent permettre de répondre aux questions du niveau de stratégie en utilisant les résultats commandés au niveau de fabrication des routes. Ce niveau traite d'idées sur la résolution des problèmes, des façons d'assembler les concepts et les algorithmes pour obtenir des méthodes de résolution.

Les trois principales questions posées par la planification sont:

1) Quelles sont les caractéristiques importantes du problème? Toutes les contraintes présentes n'ont pas une égale importance, certaines contraintes peuvent être présentes, mais en pratique, n'être presque jamais restrictives. La longueur des routes et la charge des véhicules peuvent être limitées, mais la distribution géographique des clients peut rendre l'une ou l'autre des contraintes non-restrictive. Par exemple, si les clients sont dispersés, la contrainte sur la charge ne sera pas restrictive, mais la contrainte sur la distance deviendra importante et vice versa si les clients sont regroupés.

   Quelques estimateurs peuvent être utilisés pour déterminer l'importance de chaque contrainte: le nombre moyen de clients par véhicule et le nombre minimum de véhicules nécessaires. Ces estimateurs empiriques peuvent être calculés pour chaque contrainte en tenant compte des données de l'exemplaire du problème ou de la solution obtenue. Ces estimateurs seront utiles lors des choix de la méthode de résolution.

2) Trouver une modification à la solution pour répondre à des objectifs? A l'aide de modifications mineures à la solution, celle-ci doit se conformer aux objectifs manquants, tout en demeurant réalisables pour les objectifs déjà atteints. Le système cognitif doit choisir parmi les modifications ponctuelles de la solution

(retrait, ajout, échanges de clients entre routes) lesquelles sont intéressantes.

3) Trouver une méthode de solution et la solution? Le choix d'une méthode de résolution pour un problème est déterminé par plusieurs facteurs: l'objectif, les contraintes importantes et la valeur des estimateurs. Pour un même problème, un m-TSP avec contraintes sur la capacité des véhicules, au moins deux bons algorithmes peuvent s'appliquer avec succès selon les valeurs des estimateurs.

Pour des problèmes avec peu de clients par véhicule, il est avantageux d'utiliser une approche de choix des routes par recouvrement des clients ou chaque colonne du problème de recouvrement représente une route possible. Le faible nombre de clients par route rend possible la génération efficace de colonnes.

Dans le cas où nous avons plusieurs clients par véhicule, l'approche de Fisher-Jaikumar [FISH 81] est intéressante. Les clients sont affectés aux routes par un problème d'affectation généralisée de façon à respecter la charge des véhicules tout en regroupant les clients. L'ordre de visite des clients à l'intérieur de chaque route est déterminé en appliquant un algorithme pour résoudre un problème de commis-voyageur aux clients de chaque route. D'autres algorithmes pour résoudre ce problème existent.

L'exemple du m-TSP avec contrainte sur la charge des véhicules illustre bien la multiplicité des assemblages possibles pour créer une méthode de résolution d'un problème. Un bon choix est primordial, car chacun des algorithmes exposés fonctionne mal hors de son champ de spécialisation.

De nombreux autres exemples d'assemblages de concepts et d'algorithmes pour résoudre d'autres problèmes particuliers de routing pourraient être énumérés. Il sera plus intéressant de voir quelques unes des idées de base utilisées:

1) Regrouper les clients de façon à respecter les contraintes, puis créer une route pour chaque groupe.

2) Créer une seule route avec tous les clients et la diviser en pièces respectant les contraintes.

3) Insérer les clients à une des routes selon un critère économique, tout en respectant les contraintes.

4) Echanger les clients entre les routes pour améliorer la solution, tout en conservant la réalisabilité de la solution.

Le choix d'une méthode de résolution demande la constitution d'une base de savoir assez considérable. Cette tâche pourra être facilitée par l'utilisation d'une abondante littérature sur le sujet.

La réponse à ces trois types de questions exige l'utilisation du savoir du troisième niveau: la fabrication des routes. Les algorithmes utilisés peuvent être divisés en quatre grandes catégories:

1) La fabrication de routes: cette catégorie comprend tous les algorithmes utilisés pour fabriquer une ou des routes. Elle comprend:
   - arbre de poids minimum
   - affectation
   - flots à coût minimum
   - plus court chemin (avec ou sans contraintes)
   - chaîne eulérienne
   - chemin hamiltonien (ou un bon heuristique)
   - etc

2) Algorithmes de regroupement: cette seconde catégorie comprend tous les algorithmes et concepts utilisés pour séparer des points en classes. Voici quelques éléments de cette catégorie:
   - sac alpin
   - affectation généralisée
   - regroupement par région
   - etc

3) Procédures de modification de la solution: cette troisième catégorie contient les diverses procédures de modification et d'ajustement des solutions. Les éléments les plus connus de cette catégorie sont:

   - insertion d'un client
   - retrait d'un client
   - échange de client

4) Calculs des coûts: tout algorithme d'optimisation opère ses sélections sur une fonction de coût. Cette quatrième catégorie contient les procédures calculant les fonctions de coût déterminées par le niveau de planification. Les coûts peuvent être définis de bien des façons: coût de l'ajout à une route, distance d'un centroïde fictif, pénalité, etc. La qualité des méthodes de solution dépend de la bonne définition et du bon calcul des fonctions de coûts associées à chacune des étapes. Voici quelques façons de calculer les coûts:

   - chemin le plus court
   - distance euclédienne
   - économies de Clarke & Wrigth
   - multiplicateurs de Lagrange
   - etc.

Le niveau de fabrication des routes doit aussi disposer des procédures nécessaires pour évaluer les estimateurs nécessaires pour comparer les contraintes.

4.3 Quelques détails importants

Un tel système expert permettrait à un usager de définir son problème et interactivement de le modifier jusqu'à l'obtention d'une solution acceptable. Ce dialogue reproduit, en accéléré, celui qui se déroule entre un usager et un analyste en recherche opérationnelle. Pour être viable, un tel système devra soigner le développement de deux parties primordiales du savoir: la construction d'une méthode de solution et la structure des coûts.

De nombreuses méthodes particulières existent pour résoudre chacun des problèmes de routing. Chacune de ces méthodes fut mise au point pour résoudre certains types d'exemplaires du problème et, à moins d'assurer l'optimalité, possède des points faibles. Il est important de connaître les performances de chaque méthode envisagée en fonction des estimateurs sur les contraintes des problèmes. Cette information permet de construire les règles de construction de la méthode en terme des contraintes importantes et des estimateurs de toutes les caractéristiques.

L'autre aspect important lors de la constitution de la base de savoir est de définir des structures de coût appropriées pour chaque étape de la méthode proposée. Ces structures permettent la circulation de l'information entre les différentes parties de l'algorithme. Si nous considérons à nouveau l'algorithme de Fisher-Jaikumar, la différence entre de bons et de mauvais résultats se situe au niveau de la définition des coûts lors de l'affectation généralisée. Si l'estimation du coût de joindre une route se

rapproche du coût exact, la performance de l'algorithme sera très bonne. Elle devient plutôt moyenne si les coûts estimés sont éloignés des coûts exacts.

En bref, l'important pour qu'un tel système remplisse bien son but est que sa base de connaissance dispose de bons estimateurs sur l'importance de chaque contrainte de façon à permettre la construction d'une méthode de résolution. La base de connaissance doit aussi disposer d'un certain nombre d'algorithmes classiques d'optimisation et de construction de routes. La construction de la méthode de résolution à partir des estimateurs et des algorithmes repose sur les connaissances du niveau de planification. Ces connaissances permettent la sélection des algorithmes utilisés et la construction d'une structure de coût permettant à l'information de circuler entre les étapes de la méthode proposée.

Cette courte description de quelques aspects d'un système expert exigerait, avant d'être implantée, un raffinement considérable. Elle permet, toutefois, d'illustrer une application possible d'un système expert en recherche opérationnelle.

CONCLUSION

Depuis une quinzaine d'années, les systèmes experts ont connu un développement considérable. Ils sont passés d'un gros programme à un amas de connaissances contrôlées par un interprète et, finalement, à un savoir structuré divisé en niveaux ou à une représentation mixte du savoir. Parallèlement, le développement de l'interface avec l'usager, de la capacité de justifier le raisonnement et de la possibilité d'acquérir automatiquement des connaissances se poursuivaient.

Le monde de la recherche opérationnelle est resté relativement à l'écart de tout ce développement jusqu'à maintenant. Un effort pourrait se faire selon deux axes. Premièrement, en participant au développement des mécanismes internes des systèmes experts pour en améliorer la performance.

La deuxième voie pourrait être plus facile. Depuis quelques années, plusieurs systèmes permettent le développement facile d'un système expert. Ces systèmes comportant toutes les composantes d'un S.E., sauf le savoir qu'il suffit de définir adéquatement pour obtenir un système expert dans un nouveau domaine. Cette approche pourrait être envisagée pour développer des systèmes experts en R.O. par exemple, un système sur le routing tel celui proposé dans la quatrième section.

Il est bien connu que l'un des principaux freins à une utilisation massive de la R.O. est le niveau d'expertise qu'il faut atteindre pour résoudre efficacement un problème et la difficulté d'explication d'une solution. Le développement de tels systèmes experts permettrait peut-être une utilisation plus facile et plus massive de la R.O.

BIBLIOGRAPHIE

[AIKI 83] Aikins, J.S., "Prototypical knowledge for Expert Systems", Artificial Intelligence 20 (1983), p.963-210.

[BODI 83] Bodin, L., Golden, B., Assad, A. and M. Ball, "Routing and Scheduling of Vehicles and Crews: The State of the Art", Computer and Operation Research, Vol 10, no2, (1983).

[BUCH 82] Buchanan, B.G., "New Research on Expert Systems", Machine Intelligence 10 (1982), p 269-299.

[CLAN 83] Clancey, W., "The Epistemology of a Rule-Based Expert System - a Framework for Explanation", Artificial Intelligence 20 (1983), p 215-251.

[DAVI 78] Davis, R., "Knowledge Acquisition in Ruled-Based Systems- Knowledge about Representation as a Basis for System Construction and Maintenance", in "Pattern-Directed Inference Systems" édité par D.A. Waterman et F. Hayes-Roth, Academic Press (1978) p 99-134.

[DUDA 78] Duda, R. et Al, "Semantic Network Representation in Ruled-Based Inference Systems", in Pattern-Directed Inference Systems" édité par D.A. Waterman et F. Hayes-Roth, Academic Press (1978), p.203-281.

[DUDA 81] Duda, R. et J. Gasching, "Knowledge-Based Expert Systems Come of Age", BYTE 6 (1981) p 238-281.

[ELLI 82] Elliot, R.J. et Lesk, "Route Finding in Street Map by Computer and People", Proc AAAI (1982), p 258-261.

[FISH 81] Fisher, M., and Jaikumar, R., "A generalized Assignment Heuristic for Vehicle Routing, Network, 11(2), p 109-124.

[HAYE 78] Hayes-Roth, F., Waterman, D.A. et Lenat, D.B., "Principles of Pattern-Directed Inference Systems" in Pattern-Directed Inference Systems" édité par D.A. Waterman et F. Hayes-Roth, Academic Press (1982), p 577-601.

[HAYE 79] Hayes-Roth B.et Hayes-Roth F, "A Cognitive Model of Planning", Cognitive Science 3, (1979), p 275-310.

[HAYE 83] Hayes-Roth, F., Waterman, D.A. et Lenat, D.B., "Building Expert Systems", Addison-Wesley (1983).

[LAUR 78] Laurière, J.L., "A Language and a Program for Stating and Solving Combinatorial Problems", Artificial Intelligence 10 (1978), p 29-127.

[LAUR 82A] Laurière, J.L., "Représentation et utilisation des connaissances partie 1: Les systèmes experts, ISI, vol 1, no 1, (1982), p 25-42.

[LAUR 82B] Laurière, J.L., "Représentation et utilisation des connaissances partie 2: Représentation des connaissances", TSI, vol 1, no 2, (1982), p 109-133.

[PINS 81] Pinson, S., "Représentation des connaissances dans les systèmes experts", RAIRO Informatique, vol 15, no 4 (1981), p 343-367.

[SHOR 76] Shortliffe, E., "Computer-Based Medical Consultation: MYCIN", American Elsevier (1976).

[STEF 81A] Stefik, M., "Planning with Constraint", Artificial Intelligence 16 (1981), p 111-140.

[STEF 81B] Stefik, M., "Planning and Meta-Planning", Artificial Intelligence 16 (1981), p 141-170.

[STEF 82] Stefik, M. et al, "The Organization of Expert Systems, a Tutorial", Artificial Intelligence 18 (1982) p 135-173.

[WILE 81] Wilensky, R., "Meta-Planning: Representating and Using Knowledge about Planning in Problem Solving and Natural Language Understanding", Cognitive Science 5 (1981), p 197-233.

[WINO 75] Winograd, T., "Frame Representation and the Procedural/ Declarative Controversy" in Representation and Understanding: Studies in Cognitive Science" édité par D.G. Bobrow et A. Collins, Academic Press (1975), p 185-210.